鳥取大学教育学部
研究報告（自然科学），
47（1998），13–29

選挙の当選予想プログラム

——乱数発生によるシミュレーション——

後藤　和雄，浜野　直也*

Kazuo GOTO, Naoya HAMANO: A program simulating the result of a vote

（1998年3月30日　受付）

1 はじめに

選挙の途中に開票率何%，得票数が発表される。その場合テレビなどで，いろいろシミュレーションを行っているようである。我々にもそのようなことが簡単に行えるようにならないかと考えたものが，この論文の目的である。

各開票率とその得票数に応じて，その後のシミュレーションを行ったものをグラフにしている。

開票場の票が一様にかき混ぜられていれば，このシミュレーションを繰り返すことで統計的に当選確率を求めることができる。さらに，途中で当選および当選確率をも求めている。

実際の選挙では，選挙参謀が各地区を細かく区切り，演説会や電話依頼その他，などからその地区での票を細かく読んでいる。その票を読むノウハウは非常におもしろいが，今回はこの点は触れずに，開票率と得票数が与えられたものとしてのシミュレーションを考えている。

選挙の開票ではこのようなことはしなくても，数時間後には正確に当落が判明するのであるから，どうこう言うことはないし，結果について後からいろいろと議論するのも何の意味もなさないという考えもある。しかし，このプログラムはいろいろと応用ができるものである。選挙の開票という形で分かりやすく述べているだけである。

プログラムは，木田祐司氏が開発したUBASIC [4] を用いて，作成されている。このソフトは，フリーソフトであり，多倍長計算も複素数計算もできる数学計算用ソフトである。マウスなどに対応したものではない。Visual Basicを用いたものは，今回載せていない。必要に応じて入出力部分を変更すれば，簡単に作成できるものである。

2 アルゴリズム

立候補者M人，当選者数N人，前評判（予想得票率）を入力すると，シミュレーションをする。その後，開票率と票数を入れてその時点で当選者がいれば決定し，その得票数の比率を，その後の得票確率として，100票の整数倍毎に再度得票確率を変化させるシミュレーションプロ

* Department of Mathematics, Faculty of Education, Tottori University, Tottori, City, 680–0945.

グラムである。

当選者を決定するアルゴリズム

N 人が当選するとする。得票数の多い順に並べ替えて 1 番目の人と，2 から N+1 番目の人との差の合計が，残りの票数を上まれば 1 番目の人の当選が決まる。2 番目の人は，3 から N+1 番目の人との差の合計が，残りの票数を上まれば当選が決まる。以下 N 番目まで同様の方法で判定を行う。

リードする確率としてよく知られているものに次のものがある。

投票の定理（[**1. p. 153**] [**3. p. 102**] [**2. p. 73**]）

得票の分布が一様分布と仮定し，得票するとする。このとき，投票の結果候補者 A が p 票，候補者 B が q 票を得て（$p>q$），開票して票読みをしている間，いつも A が B より多い票数を得ている確率は$\frac{p-q}{p+q}$で与えられる。

注意：全体の票数が決まっていなくて，無限に続いていると考えている。実際には，その時点での最低当選票数との差や得票率の差を考えねばならないので、これを実際適用すると思ったより小さい確率をうる。

得票数を多い順に並べて，1 番から N 番目の人と，$N+1$ 番目の人を比べて，ずっとリードする確率は上の投票の定理を用いて求められる。実際は得票は一様分布ではないので，上記の確率以上で当選する。得票の分布が一般の場合は，[2, Chap. XII] を参照せよ。

簡単に分かることであるが，2 つのことを定理としてかいておく。

Theorem 1. 全体の投票総数を A とし，当選者数は N 人とする。このとき，

$$\left[\frac{A}{N+1}\right]$$

より多くの票を獲得すると，必ず当選する。ただし，$[x]$ は x の整数部分である。

Proof. 最低当選票数が最大になるのは，$N+1$ 人が同じ得票をしたときである。すなわち，$\left[\frac{A}{N+1}\right]$ であり，これより多ければ，他の人は，その数だけ得票が少なくなるからである。よって，証明される。

Theorem 2. 立候補者を M とし，当選者を N とする。全体の投票総数を A とし，当選者が x $(1\leq x\leq N-1)$ 人すでに決まっているとする。その x 人の獲得票数の和を y とする。また，その時点で獲得票数を大きさの順に並べて，$N+2$ 番目から M 番目までの立候補者の票数の和を z とする。このとき，

$$\left[\frac{A-y-z}{N-x+1}\right]$$

より多くの票を獲得することが次の人が当選するための必要十分条件である。

Proof. Theorem 1 において，A を $A-y-z$，N を $N-x$ に置き換えることによって明らか。

注意：Theorem 1 により，最低当選ラインが求められる。また Theorem 2 により，複数当選の場合，上位当選者と下位得票の立候補者が多くの票を獲得することにより，最低当選ラインが低下することが分かる。

3 実　行　例

図 1 は，立候補者 4 名，当選者 2 名，有権者数 8 万人，投票率70%とそれぞれの前評判を入力して，シミュレーションした結果である。Theorem 1 より，18,667票以上で必ず当選する。が、1 位が20,475票を取っているのでそれにつれて当選票数は減少していることが読みとれる。

次に表 1 のように各開票率と開票数を入力して，図 2 のような結果を得る。この例では開票率80%で，候補者 2 の当選が決まる。その後，この80%での得票数に比例して得票するように乱数を発生させてシミュレーションを行うと，図 3 のようになる。

また，当選ライン上に複数の人がいて決選投票をしなければならない場合の例が図 4 に少ない票数で分かりやすく例示している。図では分かりにくいが，一番上の線は 9 票得た 2 人が重

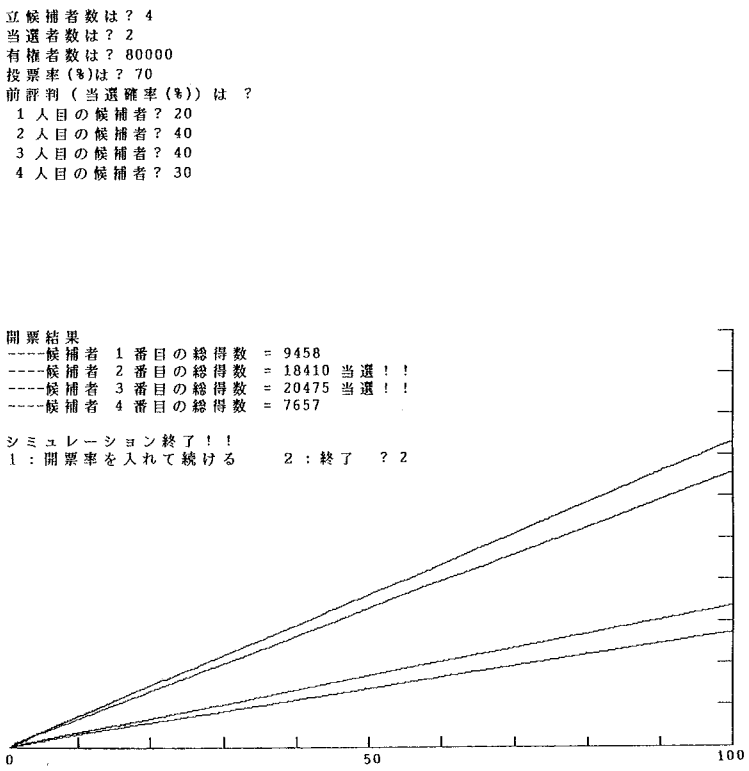

図 1

表1

開票率	10	20	30	40	50	60	70	80
候補者1番目の得票数	600	1500	1800	2000	2200	2600	3000	3200
候補者2番目の得票数	1500	3500	6000	8500	11000	13000	16000	19000
候補者3番目の得票数	2500	4000	5500	6500	7500	9000	10000	11000
候補者4番目の得票数	1000	2200	3500	5400	7300	9000	10200	11600

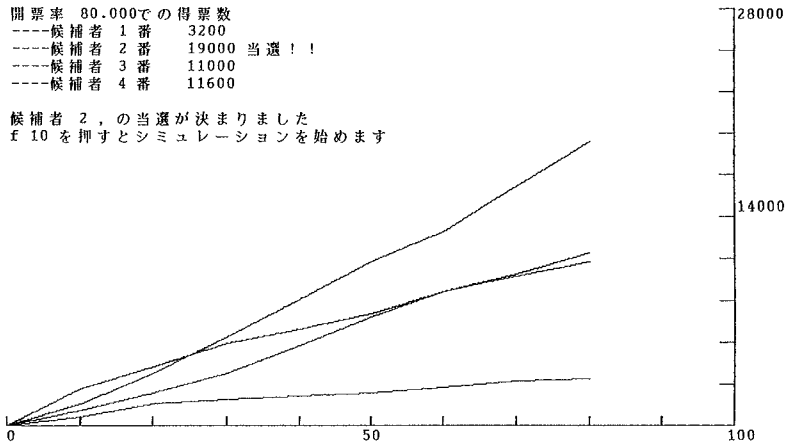

図2

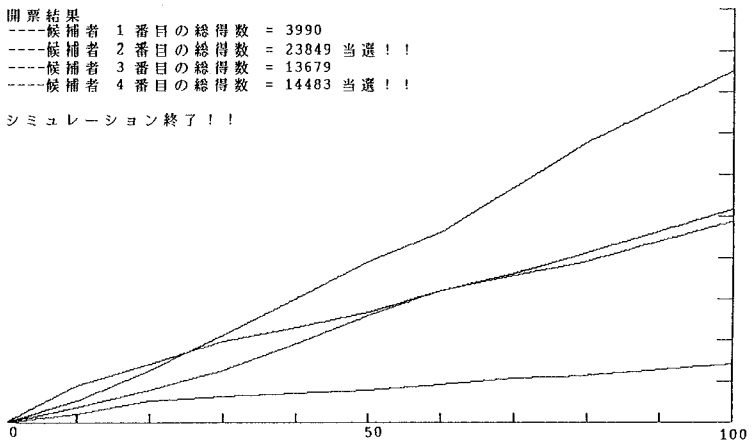

図3

なり，上から3番目の線は7票得た2人が重なっている。つまり，当選者は4名だが，その4番目に得票数の多い人が7票で重なってしまったためこの2人による決選投票をしなければならないことになる。このような決戦投票になることも考慮したプログラムである。

```
立候補者数は？ 8
当選者数は？ 4
有権者数は？ 50
投票率(%)は？ 100
前評判（当選確率(%)）は ？
 1 人目の候補者？ 10
 2 人目の候補者？ 10
 3 人目の候補者？ 10
 4 人目の候補者？ 10
 5 人目の候補者？ 10
 6 人目の候補者？ 10
 7 人目の候補者？ 10
 8 人目の候補者？ 10
```

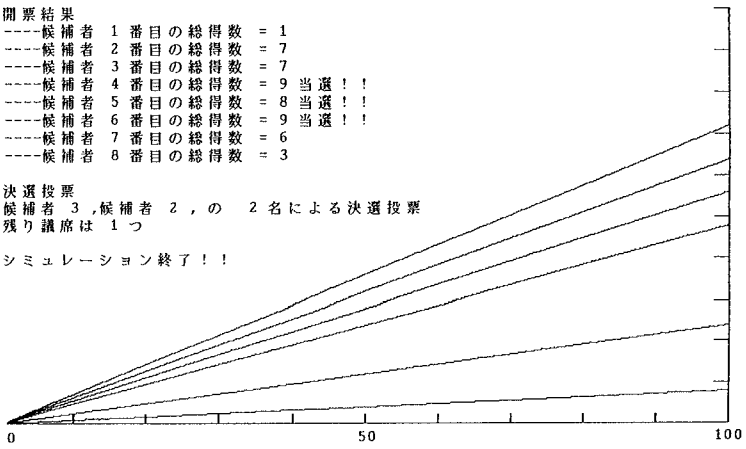

図 4

4 考　　察

今回のプログラムは途中の開票情報により推測される得票の伸びを考慮したプログラムではない。伸びが止まるといったことが起こるのは開票場所における偏りに起因する。実際，各開票場における投票総数と開票結果に偏りが開票場では起こるからである。このプログラムでこの点を考慮するには，各開票場を別々の対象と想定し，プログラムを実行し，合計すれば対応可能である。今後，機会があれば発表したい。Sequential statistical test を用いて，当選の統計的または確率的判定を行うこともできるが，この論文ではそのようなものを含んだものになっていないことが今後の改良点である。また，実際の選挙の開票に適応して，充分使えるシステムであることを確認している。以上述べた，改良点は subroutine を付け加えることで簡単に作成できるので，自由に変更して使ってもらいたい。

Abstract

We give a algorithm and a UBASIC program which simulates the result of a vote using random number. We treat also the case of a tie and show some examples.

参 考 文 献

[1] 武隈良一，現代数学レクチャーズ A-3，確率，培風館，1978 (1993年第14刷)．
[2] W. Feller, An introduction to probability theory and its applications, Vol. 1 (3rd ed., 1968), Vol. II, Chap.XII.
[3] フェラー，河田竜夫監訳，確率論とその応用 I 上，1960.
[4] 木田祐司：UBASIC 86，第8.7版 ユーザーズマニュアル，日本評論社.

5 プログラムの解説

このプログラムは，BASIC で最近使える関数サブルーチンを用いていない。そのため，N88BASIC にもすぐにかき直せる。また，関数サブルーチンを使ったプログラムにも容易にかき直すことができる。

以下に，プログラムに対応する説明をかく。

設定 1 (*settei1) (1100－1170) 以下 ( ) 内は行番号を表す
　立候補者数を M に
　当選者数を N に入力する。
　不適の場合はやり直し。
設定 2 (*settei2) (1190－1250)
　有権者数を A に
　投票率 (%) を B に入力する。
　不適の場合はやり直し。
H＝100 と確率変動の間隔を定義 (1270)
R＝10 と開票率を入れて続ける回数の上限を定義 (1280)
前評判 (*settei3) (1310－1450)
　候補者 K の前評判 (%) を X(K) に入力する。(K＝1, ..., M)
　不適の場合はやり直し。
　X(K) に，全体での百分率に直したものを入れ直す。
グラフの初期設定 (1500－1680)
　D＝2*C/M と定義。これは，グラフの縦座標の上限である。
　横軸，縦軸，目盛りを表示する。
開票前設定 (1750－1850)
　Xx(M)　　候補者 K の得票数を入れる配列を定義。
　S(M), T(M)　候補者 K のグラフの座標を入れる配列を定義。
　Jun(M, 2)　候補者 K の順位を入れる配列を定義。
　Hante(M)　候補者 K が当選しているかどうか判定する配列を定義。

初期値は０。当選が決まると１を入れる。

Kai(10, M)　開票率を入れ直してもう１度シミュレーションするときに，候補者Ｋのそのときの得票数を入れる配列を定義。

I=1　開票数。

J=1　確率変動の回数。

開票（*Kaihyou）（1900－2040）

Ｔに０から１の疑似乱数を入れそれに対応する候補者に票を入れていく。

この作業を繰り返す。

開票が終了するか，確率変動する開票数に達したとき次の過程（*Totyu）にうつる。

ソート処理（*Totyu）（2090－2180）

Jun(K, 1）に候補者Ｋの番号を入れる。

Jun(K, 2）に候補者Ｋの現時点の得票数を入れる。

Jun(K, 1..2）を得票数の多い順に並べ替える。

確率変動（2200－2230）

もし，開票が終了していれば，開票終了（*Syuryo1）にうつる。

X(K）に候補者Ｋの現時点での人気率を入れる。

判定（2260－2330）

候補者Ｋが当選を決めていれば，Hante(K）に１を入れる。

もしZ=1（再シミュレーション開始のとき）ならば，当選判定に戻る。

グラフ等の表示（2370－2590）

現時点の開票率，候補者Ｋの得票数と，当選か否かを表示する。

現時点までの得票数の移り変わりをグラフ表示する。

開票（*Kaihyou）に戻る（2600）

開票終了（*Syuryo1）（2650－2700）

得票数の多かった上位Ｎ人が当選するので，それらの人にHante(K）に１をいれる。

もし，得票数がＮ番目の人とＮ+1番目の人の得票数が同じときは，決選投票(*Kessen1）へうつる。

そうでないなら，終了表示（*Syuryo2）へうつる。

決選投票（*Kessen1, *Kessen2）（2730－2910）

得票数がＮ番目の人と同じ人を調べる。

終了表示（*Syuryo2）（2950－3100）

候補者Ｋの得票数と，当選か否かを表示する。

得票数の移り変わりをグラフ表示する。もし，得票数がＮ番目の人とＮ+1番目の人の得票数が同じときは，決選投票の表示（*Kessen3）へうつる。

そうでないなら，シミュレーション終了（*Syuryo3）へうつる。

決選投票の表示（*Kessen3）（3120－3220）

残り議席と決選投票される候補者を表示する。

シミュレーション終了（*Syuryo3）（3260－3400）

再シミュレーションの回数が10回に達していれば，終了（*Owari）へうつる。

開票率を入れて，もう１度シミュレーションしたいときは，再選挙（*Tuduki）へうつる。

終了したいときは，終了（*Owari）へうつる。
再シミュレーション（*Tuduki）（3450－3460）
開票率（*settei5）（3480－3520）
現在の開票率を入力する。
不適の場合はやり直し。
得票数（*settei6）（3540－3710）
候補者Kの現在の得票数をKai(Y, K）に入力する。
Yは，再シミュレーションした回数＋1である。
不適の場合はやり直し。
グラフの再設定（3760－3970）
横軸，縦軸，目盛りを表示する。
現時点までの得票数の移り変わりをグラフ表示する。
初期値の入れ直し（4010－4060）
当選判定（4080－4090）
開票率を入れた時点で，当選者が出るか判定する。
Z＝1の値を持って，*Totyuへうつる。
グラフ等の再表示（4130－4410）
現在の開票率までの候補者Kの得票数と，当選か否かを表示する。
現在の開票率までの得票数の移り変わりをグラフ表示する。
プログラムを一時停止する。f10を押せば続けることができる。
初期値の入れ直し（4460－4530）
終了（*Owari）（4580－4590）

6 プログラム

```
1000  '----------------------------------------------------------------------------------------------------
1010  ' 選挙に関するシミュレーション
1020  ' Programed by UBASIC
1030  ' DOS/V で使うには次のように2960行をかき直し，
1040  ' 2965行を付け加えると，使いやすくなります．
1050  ' 2960 locate 0,0:print "                                                                    "
1060  ' 2965 locate 0,0
1070  '----------------------------------------------------------------------------------------------------
1080  '
1090  '初期設定
1100  *Settei1
1110  cls 3
1120  locate 0,0
1130  print "立候補者数は";
1140  input M
```

```
1150  print "当選者数は";
1160  input N
1170  if M<=0 or N<=0 or N>=M then print "ERROR!!":goto *Settei1
1180  '
1190  *Settei2
1200  print "有権者数は";
1210  input A
1220  print "投票率（%）は";
1230  input B
1240  if A<=0 or B<=0 or B>100 then print "ERROR!!":goto *Settei2
1250  C=int（A*B/100）
1260  '
1270  H=100 '確率変動の間隔
1280  R=10 '開票率を入れる回数の上限
1290  '
1300  '前評判
1310  dim X(M)
1320  S=0
1330  print "前評判（当選確率（%））は?"
1340  K=1
1350  while K<=M
1360      *Settei3
1370      print K;"人目の候補者";
1380      input X(K)
1390      if X(K)<=0 or X(K)>=100 then print "ERROR!!":goto *Settei3
1400      S=S+X(K)
1410      K=K+1
1420  wend
1430  for K=1 to M
1440      X(K)=X(K)/S
1450  next
1460  '
1470  '----------------------------------------------------------------------------------------------------
1480  '
1490  'グラフの初期設定
1500  screen 1
1510  console 0,25,0
1520  cls 3
1530  D=2*C/M
1540  view（0,0)-(590,380)
```

```
1550    window (0,-D)-(100,0)
1560    line (0,0)-(100,0),7
1570    line (100,0)-(100,-D)
1580    for I=0 to 100 step 10
1590        line (I,0)-(I,-D/50)
1600    next
1610    for I=0 to 10
1620        line (100,-I*D/10)-(98,-I*D/10)
1630    next
1640    locate 0,32:print "0";
1650    locate 36,32:print "50";
1660    locate 73,32:print "100";
1670    locate 73,0:print int(D);
1680    locate 73,11:print int(D/2);
1690    '
1700    '----------------------------------------------------------------------------------------------------------------
1710    '
1720    '開票前設定
1730    randomize
1740    '
1750    dim Xx(M)
1760    dim S(M)
1770    dim T(M)
1780    dim   Jun(M,2)
1790    dim   Hante(M)
1800    dim   Kai(R,M)
1810    '
1820    I=1
1830    J=1
1840    X=1
1850    Y=2
1860    '
1870    '----------------------------------------------------------------------------------------------------------------
1880    '
1890    '開票
1900    *Kaihyou
1910    while (I<=C)
1920        T=rnd
1930        S=0
1940        K=1
```

```
1950        while (K<=M)
1960            S=S+X(K)
1970            if S>=T then goto *A
1980            K=K+1
1990        wend
2000        *A
2010        Xx(K)=Xx(K)+1
2020        if I=C or I=H*J then goto *Totyu
2030        I=I+1
2040    wend
2050    '
2060    '----------------------------------------------------------------------------------------------------
2070    '
2080    'ソート処理と確率変動
2090    *Totyu
2100    for K=1 to M
2110        Jun(K,1)=K
2120        Jun(K,2)=Xx(K)
2130    next
2140    for K=1 to M-1
2150        for L=K+1 to M
2160            if Jun(K,2)<Jun(L,2) then swap block Jun(K,1..2),Jun(L,1..2)
2170        next
2180    next
2190    '
2200    if I=C then goto *Syuryo1
2210    for K=1 to M
2220        X(K)=Xx(K)/(H*J)
2230    next
2240    '
2250    '判定
2260    *Hantei
2270    if Hante(Jun(X,1))=1 then X=X+1
2280    SS=0
2290    for K=X+1 to N+1
2300        SS=SS+Xx(Jun(X,1))-Xx(Jun(K,1))
2310    next
2320    if SS>C-I then Hante(Jun(X,1))=1:goto *Hantei
2330    if Z=1 then return
2340    '
```

```
2350 '----------
2360 '
2370 locate 0,0
2380 print "開票率";:print using(3,3),100*H*J/C;:print "での人気"
2390 for K=1 to M
2400     if Hante(K)=0 then goto *T1
2410     color K@6+1:print "--------";
2420     color 7:print "候補者";K;"番　";
2430     print using(1,4),Xx(K)/(H*J);
2440     color 4:print "当選！！":color 7:goto *Tugi1
2450     *T1
2460     color K@6+1:print "--------";
2470     color 7:print "候補者";K;"番　";
2480     print using(1,4),Xx(K)/(H*J)
2490     *Tugi1
2500 next
2510 '
2520 for K=1 to M
2530     line (S(K),-T(K))-(100*H*J/C,-Xx(K)),K@6+1
2540     S(K)=100*H*J/C
2550     T(K)=Xx(K)
2560 next
2570 '
2580 J=J+1
2590 I=I+1
2600 goto *Kaihyou
2610 '
2620 '----------
2630 '
2640 '開票終了
2650 *Syuryo1
2660 for K=1 to N
2670     Hante(Jun(K,1))=1
2680 next
2690 if Xx(Jun(N,1))=Xx(Jun(N+1,1)) then goto *Kessen1
2700 goto *Syuryo2
2710 '
2720 '決選投票
2730 *Kessen1
2740 K=1
```

```
2750  Kk=1
2760  Hante(Jun(N,1))=0
2770  while K<N
2780      if Xx(Jun(N,1))<>Xx(Jun(N-K,1))then goto *Kessen2
2790      Hante(Jun(N-K,1))=0
2800      K=K+1
2810      Kk=Kk+1
2820  wend
2830  '
2840  *Kessen2
2850  K=2
2860  Kkk=1
2870  while K<M-N
2880      if Xx(Jun(N,1))<>Xx(Jun(N+K,1))  then goto *Syuryo2
2890      K=K+1
2900      Kkk=Kkk+1
2910  wend
2920  '
2930  '----------------------------------------------------------------------------------------------------------------
2940  '
2950  *Syuryo2
2960  cls 1 'DOS/V は，ここをかき直してください
2970  print "開票結果"
2980  for I=1 to M
2990      if Hante(I)=0 then goto *T2
3000      color I@6+1:print "--------";
3010      color 7:print "候補者";I;"番目の総得数=";Xx(I) ;
3020      color 4:print "当選！！":color 7:goto *Tugi2
3030      *T2
3040      color I@6+1:print "--------";
3050    color 7:print "候補者";I;"番目の総得数=";Xx(I)
3060    *Tugi2
3070  next
3080  '
3090  if Xx(Jun(N,1))=Xx(Jun(N+1,1))  then goto *Kessen3
3100  goto *Syuryo3
3110  '
3120  *Kessen3
3130  print
3140  print "決選投票"
```

```
3150  for K=1 to Kk
3160      print "候補者";Jun(N-K+1,1);",";
3170  next
3180  for K=1 to Kkk
3190      print "候補者";Jun(N+K,1);",";
3200  next
3210  print " の ";Kk+Kkk;"名による決選投票"
3220  print "残り議席は";Kk;"つ"
3230  '
3240  '-----------------------------------------------------------------------------------------------------------------
3250  '
3260  *Syuryo3
3270  for K=1 to M
3280      line (S(K),-T(K))-(100,-Xx(K)),K@6+1
3290  next
3300  '
3310  print
3320  print "シミュレーション終了！！"
3330  if Y=R goto *Owari
3340  '
3350  *Settei4
3360  print " 1 ：開票率を入れて続ける      2 ：終了  ";
3370  input Q
3380  if Q=1 then goto *Tuduki
3390  if Q=2 then goto *Owari
3400  print "ERROR!!":goto *Settei4
3410  '
3420  '-----------------------------------------------------------------------------------------------------------------
3430  '
3440  '再シミュレーション
3450  *Tuduki
3460  cls 1
3470  '
3480  *Settei5
3490  print "現在の開票率（%）は？  ";
3500  input U
3510  if U<=Kai(Y-1,0)  or U>=100 then print "ERROR!!":goto *Settei5
3520  Kai(Y,0)=U
3530  '
3540  *Settei6
```

```
3550  print "開票率";U;"%での得票数は ?"
3560  print "開票数は";int(C*U/100);"票です"
3570  print "誤差は";int(C*U*5/10000);"票以内でお願いします"
3580  '
3590  K=1
3600  S=0
3610  while K<=M
3620      *Settei7
3630      print K;"人目の候補者";
3640      input Kai(Y,K)
3650      if Kai(Y-1,K)>Kai(Y,K)  then print "ERROR!!":goto *Settei7
3660      Xx(K)=Kai(Y,K)
3670      S=S+Xx(K)
3680      K=K+1
3690  wend
3700  '
3710  if abs(S-C*U/100)>C*U*5/10000 then print "ERROR!!":goto *Settei6
3720  '
3730  '----------------------------------------------------------------------------------------------------------------
3740  '
3750  'グラフの再設定
3760  cls 3
3770  view (0,0)-(590,380)
3780  window (0,-D)-(100,0)
3790  line (0,0)-(100,0) ,7
3800  line (100,0)-(100,-D)
3810  for I=0 to 100 step 10
3820      line (I,0)-(I,-D/50)
3830  next
3840  for I=0 to 10
3850      line (100,-I*D/10)-(98,-I*D/10)
3860  next
3870  locate 0,32:print "0";
3880  locate 36,32:print "50";
3890  locate 73,32:print "100";
3900  locate 73,0:print int(D) ;
3910  locate 73,11:print int(D/2) ;
3920  '
3930  for K=1 to Y
3940      for L=1 to M
```

```
3950            line (Kai(K-1,0),-Kai(K-1,L))-(Kai(K,0),-Kai(K,L)),L@6+1
3960        next
3970    next
3980    '
3990    '-----------------------------------------------------------------------------------------------------------
4000    '
4010    I=S
4020    X=1
4030    Z=1
4040    for K=1 to M
4050        Hante (K)=0
4060    next
4070    '
4080    gosub *Totyu
4090    Z=0
4100    '
4110    '-----------------------------------------------------------------------------------------------------------
4120    '
4130    locate 0,0
4140    print "開票率";:print using(3,3),U;:print "での得票数"
4150    for K=1 to M
4160        if Hante(K)=0    then goto *T3
4170        color K@6+1:print "--------";
4180        color 7:print "候補者";K;"番　";
4190        print Xx(K);
4200        color 4:print "当選！！":color 7:goto *Tugi3
4210        *T3
4220        color K@6+1:print "--------";
4230        color 7:print "候補者";K;"番　";
4240        print Xx(K)
4250    *Tugi3
4260    next
4270    '
4280    print
4290    if Hante(Jun(1,1))=1 then goto *G
4300    print "当選者はまだいません"
4310    goto *Tugi4
4320    '
4330    *G
4340    for K=1 to X-1
```

```
4350        print "候補者";Jun(K,1);", ";
4360    next
4370    print "の当選が決まりました"
4380    '
4390    *Tugi4
4400        print "f10 を押すとシミュレーションを始めます"
4410    stop
4420    '
4430    '----------------------------------------------------------------------------------------------------------------------------
4440    '
4450    '初期値の再設定
4460    Y=Y+1
4470    J=int(S/H)+1
4480    for K=1 to M
4490        S(K)=U
4500        T(K)=Xx(K)
4510        X(K)=Xx(K)/S
4520    next
4530    goto *Kaihyou
4540    '
4550    '----------------------------------------------------------------------------------------------------------------------------
4560    '
4570    '終了
4580    *Owari
4590    end
```